Prognostic of RUL based on Echo State Network Optimized by Artificial Bee Colony

Edgar J. Amaya[1], Alberto J. Alvares[2]

[1,2]*University of Brasilia (UnB), Mechanical and Mechatronic Department, Brasilia, DF, CEP 70910-900, Brazil*

eamaya@unb.br
alvares@alvarestech.com

ABSTRACT

Prognostic is an engineering technique used to predict the future health state or behavior of an equipment or system. In this work, a data-driven hybrid approach for prognostic is presented. The approach based on Echo State Network (ESN) and Artificial Bee Colony (ABC) algorithm is used to predict machine's Remaining Useful Life (RUL). ESN is a new paradigm that establishes a large space dynamic reservoir to replace the hidden layer of Recurrent Neural Network (RNN). Through the application of ESN is possible to overcome the shortcomings of complicated computing and difficulties in determining the network topology of traditional RNN. This approach describes the ABC algorithm as a tool to set the ESN with optimal parameters. Historical data collected from sensors are used to train and test the proposed hybrid approach in order to estimate the RUL. To evaluate the proposed approach, a case study was carried out using turbofan engine signals show that the proposed method can achieve a good collected from physical sensors (temperature, pressure, speed, fuel flow, etc.). The experimental results using the engine data from NASA Ames Prognostics Data Repository RUL estimation precision. The performance of this model was compared using prognostic metrics with the approaches that use the same dataset. Therefore, the ESN-ABC approach is very promising in the field of prognostics of the RUL.

1. INTRODUCTION

Unexpected machine failures often result in production downtime, delayed delivery schedule, poor customer satisfaction, economic losses and safety issues. Condition monitoring, diagnostic and prognostic utilizes sensors signals to assess the machine's health status and make inferences about the Remaining Useful Life (RUL) (Heng, Zhang, Tan, and Mathew, 2009). The RUL at time instant t_p (time of prognostic) is calculated as a difference between the End of Life (EoL) at time instant t_p, and the actual time t_p (Shankar, 2015). RUL prognostic is a key task of a Prognostic and Health Management (PHM) system (Dong & He, 2007; Pecht, 2008; Pecht & Jaai, 2010; Gasperin, Juricic, Baskoski, and Jozef, 2011) and Condition Based Maintenance (CBM) (Wang & Zhang, 2008).

Generally, three main prognostic approaches can be distinguished (Vachtsevanos, 2006): model-based (Zhang, Zhao, Liu, Zhang, Jia, and Feng, 2011, Compare & Zio, 2014, Daroogheh, Meskin, and Khorasani, 2014, Weiming, Bing, Min, and Houjun, 2014), data-driven (Hu, Youn, and Wang, 2011, Ferreiro, Arnaiz, Sierra, and Irigoien, 2012, Li, Wang, and Ismail, 2013, Pla, Lopez, Gay, and Pous, 2013) and hybrid method (combination of model-based and data-driven) (Kumar, Torres, Chan, and Pecht, 2008, Liao & Kottig, 2014). The main advantage of model-based approach is the ability to incorporate physical understanding of the system; on the other hand, the drawback is the difficult to find mathematical representation of complex systems. The strength of data-driven techniques is their ability to transform high-dimensional data into lower dimensional information for prognostic; the main disadvantage is the high computational cost. The data-driven approach is recommendable to systems with large historical data, and where is not comprehensive their physical model and failure mechanisms.

Prognostic approaches for CMAPSS datasets was classified in the three categories by Ramasso and Saxena (2014), the first category (mapping between set of inputs and RUL) was applied in this paper. For this category is showed the following methods: RNN, EKF, MLP, RBF, KF, ANN, ESN, Fuzzy Rules, GA, used by different authors. The list of methods show that Artificial Neural Network (ANN) is one of the most used technique on data-driven approach that aims to estimate the machine's RUL processing information of machine's operational condition. From different kind of ANN, the Recurrent Neural Networks (RNN) is a powerful tool that integrates large dynamic memory and high adaptable computational capabilities. However, their training process is inherently difficult (Jaeger, 2002b, Lukosevicius & Jaeger, 2009). In the last years, Reservoir Computing (RC) concept was introduced to make RNN training very simple, producing other paradigms like Echo States Network (ESN) proposed by Jaeger (2001). An ESN has three layers: input, reservoir and readout (Jaeger, 2001), the dynamic reservoir neurons are randomly connected and their weights fixed before the training process. Then, it needs only one-step linear training for readouts. ESN arises as a solution for two characteristics often adverse: simplicity of the mathematical model and ability to approximate nonlinear dynamic behavior (Boccato, 2013).

The dynamic reservoir is the main element in the ESN, adjust their parameters with optimum values is a challenge. Evolutionary methods for pre–train the reservoir is a natural strategy in the search the best parameters and weight values (Lukosevicius & Jaeger, 2009). Several evolutionary approaches to ESN reservoir optimization have been presented (Bush & Tsendjav, 2005, Ishii, van der Zant, Becanovic, and Ploger, 2004).

Comparing with other techniques, Karaboga and Akay (2009) contrasted the ABC algorithm performance, with the performance of techniques like Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Differential Evolution Algorithm (DEA). The results shown that the ABC performance is better or similar to the other algorithms. The results obtained by Turanoglu, Ozceylan, and Kiran (2011), Butani, Gajjar, and Thakker (2011) and Hossain and El-Shafie (2014), where was compared the efficiency of the ABC and PSO algorithm, shown that the ABC is more efficient for optimal solution searching. In this work is presented an approach for ESN design using the ABC algorithm. The strategy adopted in this work, first search the ESN best parameters, after that, the reservoir weights is generated. The experiments were implemented in Matlab.

This paper is organized as follows. Section 2 outlines some basic concepts about ESN used in this paper for RUL estimation; Section 3 reports the ABC algorithm that is used to adjust optimized ESN parameters. Section 4 describes the ESN-ABC approach as a prognostic tool proposed in this work; Section 5 presents the experimental case study of turbofan engine used for validating the prognostic approach; Section 6 presents the results and discusses.

2. ECHO STATE NETWORK

In the last years, new training approach of RNN attracted attention from researchers. These methods were proposed independently by the name of Liquid State Machines (LSM) (Maass, Natschlager, and Markram, 2002, Natschlager, Maass, and Markram, 2002) and Echo State Networks (ESN) (Jaeger, 2001, Jaeger, 2002b, Jaeger & Haas, 2004). The LSM and ESN with the most recent method called Backpropagation Decorrelation (BPDC) (Steil, 2004), gave rise to the term Reservoir Computing (RC) (Verstraeten, Schrauwen, D'Haene, and Strooband, 2007, Schrauwen, Verstraeten, and Campenhout, 2007).

A pioneer RC method is the ESN, proposed by Jaeger (2001) as a RNN topology. This network includes interconnected recurrent neurons called dynamic reservoir in their hidden layer. The main characteristic of ESN is that only the readout needs to be trained while the reservoir and input weights are remained untrained. This characteristic reduces the complexity of training RNN to a simple linear regression.

2.1. Basic Structure of an ESN

The generic structure of ESN has three layers: input, hidden (reservoir) and output (readout) as shown in the Figure 1. The input layer receives information from the environment. The Dynamic Reservoir (DR), with recurrence inside, consists of a large number of neurons, usually about 20 ~ 500 neurons (Song, Zhao, Feng, An, and Song, 2011). The weight values in the DR are generated randomly. ESN exploits the dynamic of large DR to extract interesting properties of input sequence (Jaeger, 2001).

As shown in the Figure 1 the ESN has K input, L output and N hidden neurons. The weights of the connections from the input layer to DR, within DR, from DR to readout and reversely from the readout to DR is denoted as W^{in}, W, W^{out} and W^{back}, with sizes N×K, N×N, L×N and N×L respectively. The values of W^{in}, W and W^{back} are assigned randomly. However, to ensure Echo State Property (ESP) and the richness of DR behaviors, the reservoir connectivity should be sparse, and the spectral radius of W should be smaller than the unit (Jaeger, 2002b).The connectivity density and the spectral radius are denoted as d and sr, respectively. Only the weight matrix W^{out} needs to be train.

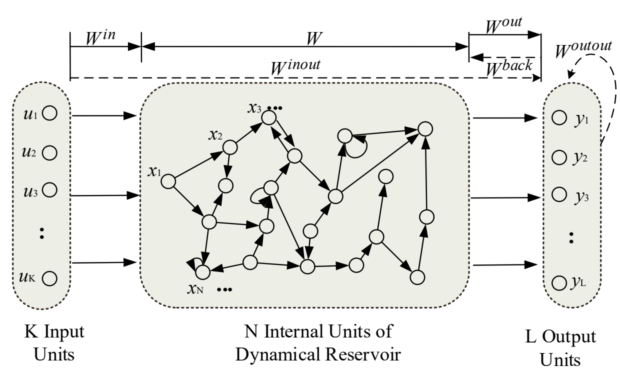

Figure 1. Basic Structure of ESN

2.2. Echo State Property (ESP)

The ESP is a basic, necessary property for the ESN learning principle to work. Under certain conditions, the reservoir states $X(n)$ becomes asymptotically independent of initial conditions and depends only on input history $U(n)$. Here $X(n)=\{x_1(n), x_2(n),\ldots, x_N(n)\}$ $x_i(n)$ is the i^{th} internal state of the neuron at time n, the input history $U(n)$ = {u(n), $u(n-1)$,$u(n-2)$,...}. In other word, it means that there exists such a function Echo, which satisfies $X(n)$ = Echo(U(n)). Metaphorically speaking, the reservoir state $X(n)$ can be considered as the so-called echo reflection of its previous inputs (Jaeger, 2001, Jaeger, 2002b).

2.3. Learning Scheme

The basic idea of ESN is to use a huge DR as a source of dynamic behavior, which neural activities are combined into desired output. ESN presents a kind of fast, simple and constructive algorithm for supervised learning of RNN. The reservoir state and the readout are updated through the Eq. (1) and Eq. (2), respectively.

$$x(n+1) = f\left(W^{in}u(n+1) + Wx(n)\right) \quad (1)$$

$$y(n) = f^{out}\left(W^{out}x(n)\right) \quad (2)$$

Where: f and f^{out} are the activation function of the reservoir neurons and readout neurons, respectively. W^{in} is the input weights, W^{out} the output weights, W the reservoir weights. $x(n)$ are the internal state of the reservoir neurons, $y(n)$ the ESN output value.

ESN training usually are considered as some linear regression problems, which can be solved via two kind of algorithms: online and offline. The ESN online algorithm can be realized via Recursive Least Square (RLS) algorithm (Jaeger, 2003). Given a T training input/output sequence <$u(1)$, $y_{target}(1)$), ..., ($u(T)$, $y_{target}(T)$>, and desire to obtain a trained ESN (W^{in}, W, W^{back}, W^{out}) whose output $y(n)$ approximates the teacher output $y_{target}(n)$, when the ESN is driven by the training input $u(n)$. The ESN offline algorithm used in this work is usually carried out by four steps: (1) Define an untrained ESN (W^{in}, W, W^{back}) which satisfies the echo state property; (2) Sample network training dynamics, drive the network by the training data; (3) Compute output weights (W^{out}); and (4) Exploitation, the network (W^{in}, W, W^{back}, W^{out}) is now ready for use. It can be driven by novel input sequences $u(n)$, using the update Eq. (1) and Eq. (2). Major details of the learning procedure are explained by Jaeger (2002b). The learning algorithm is applied in order to reduce the Mean Square Error (MSE) between the target values (y_{target}) and the ESN output (y) shown in the Eq. (3).

$$MSE = \frac{1}{T}\sum_{n=1}^{T}\left(y_{target}(n) - y(n)\right)^2 \quad (3)$$

Where: T the number of samples in the data time series of input/output used for training. The goal is to find the best W^{out} weights matrix corresponding to the lowest MSE possible result, achieved by a linear regression.

3. ARTIFICIAL BEE COLONY ALGORITHM

Artificial Bee Colony (ABC) algorithm is a swarm intelligence method which simulates intelligent behavior of honey bees. The first studies of ABC algorithm are testing the performance of the algorithm for constrained and unconstrained problems and comparing with those of other well-known modern heuristic algorithms such as Genetic Algorithm (GA), Differential Evolution (DE), Particle Swarm Optimization (PSO) (Karaboga & Basturk, 2007). The classification performance of the ABC algorithm is tested on training neural networks (Karaboga & Ozturk, 2009) and on clustering (Karaboga & Ozturk, 2010) with benchmark classification problems and the results are compared with those of other widely-used techniques. The model of ABC algorithm consist of three groups of bees; employed bees, onlooker bees and scout bees in the colony of artificial bees (Karaboga, 2010).

In order to understand the ABC algorithm is presented a parallelism with classical Genetic Algorithm (GA). The number of food source is equivalent to the number chromosomes, the nectar amount is equivalent to the fitness value, and the onlooker bee phase is similar to the crossover, the scout bee phase similar to the mutation.

Half of the colony consists of employed bees, and the other half includes onlooker bees. Employed bees are responsible for exploiting the nectar sources explored before and giving information to the waiting bees (onlooker bees) in the hive about the quality of the food source sites which they are exploiting. Onlooker bees wait in the hive and decide on a food source to exploit based on the information shared by the employed bees. Scouts either randomly search the environment in order to find a new food source depending on an internal motivation or based on possible external clues.

In ABC algorithm, the location of a food source represents a potential solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The nectar amount of a food source corresponds to the profitability (fitness) of the associated solution. Each food source is exploited by only one employed bee. In other words, the number of employed bees is equal to the number of food sources existing around the hive (number of solutions). The employed bee whose food source has been abandoned becomes a scout. Using the analogy between emergent intelligence in foraging of bees and the ABC algorithm, the units of the basic ABC algorithm can be explained as follows:

3.1. Initialization Phase

In the beginning, the ABC algorithm generates a uniformly distributed population of SN solutions (Solution Number) where each solution (i = 1, 2,..., SN) is a D-dimensional vector. Here D is the number of variables in the optimization problem and x_i represents the i^{th} food source in the population. Initial food sources are produced randomly within the range of the boundaries of the parameters, described by the Eq. (4).

$$x_{ij} = x_j^{min} + rand(0,1)\left(x_j^{max} - x_j^{min}\right) \quad (4)$$

Where: $i = 1\ldots SN$, $j = 1\ldots D$. SN is the number of food sources and D is the number of optimized parameters.

In addition, the ABC algorithm depends on the three control parameters: the first one is the population size that determines the number of food sources in population. The second is the maximum cycle number that determines the maximum number of generations. The last one is the Limit that is used to determine the number of allowable generations after which each non improved food source is to be abandoned. After producing food sources and assigning them to the employed bees, the objective function specifically for the optimization problem is operated, its value is obtained, and all the fitness values of the food sources are calculated by the Eq (5).

$$fitness = \begin{cases} 1/(1+f_i) & f_i \geq 0 \\ 1 + abs(f_i) & f_i < 0 \end{cases} \quad (5)$$

Where: f_i is the cost value of the solution. For maximization problems, the cost function can be directly used as a fitness function.

3.2. Employed Bees Phase

As mentioned earlier, each employed bee is associated with only one food source site. Hence, the number of food source sites is equal to the number of employed bees. An employed bee produces a modification of the position of the food source (solution) in her memory depending on local information (visual information) and finds a neighboring food source, and then evaluates its quality. In ABC, finding a neighboring food source is defined by Eq. (6).

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right) \quad (6)$$

Within the neighborhood of every food source site represented by x_i, a food source v_i is determined by changing one parameter of x_i, j is a random integer in the range [1, D] and $k \in \{1, 2, \ldots N\}$ is a randomly chosen index that has to be different from i. ϕ_{ij} is a uniformly distributed real random number in the range [-1, 1].

3.3. Onlooker Bees Phase

After all employed bees complete their searches, they share their information related to the nectar amounts and the positions of their sources with the onlooker bees in the dance area. This is the multiple interactive feature of the artificial bees of ABC. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source site with a probability related to its nectar amount. This probabilistic selection depends on the fitness values of the solutions in the population given by the Eq. (7).

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \quad (7)$$

In this probabilistic selection scheme, as the nectar amount of food sources (the fitness of solutions) increases, the number of onlookers visiting them increases, too. This is the positive feedback feature of ABC.

3.4. Scout Bees Phase

An employed bee become scout bee if the employed bee is associated with an abandoned food source, also, the food source is replaced by randomly choosing another food source from the search space. The scout bees phase is started when a position of a food source is not updated for a predetermined number of cycles, then the food source is assumed to be abandoned. Inside ABC, the predetermined number of cycles is a crucial control parameter, which is called limit for abandonment.

4. ESN-ABC AS A PROGNOSTIC TOOL

In this section is detailed the ESN-ABC approach, describing the motivation, the proposed approach and the RUL prognostic algorithm.

4.1. Motivation

From various optimization techniques, the ABC algorithm is highlighted by their efficiency in searching optimal solutions for different kind of problems, with the advantage of using few control parameters. These characteristics and results obtained when compared with other optimization techniques motivated the application of the ABC algorithm to adjust the ESN parameters and the application to RUL prognostic.

The main parameters that determine the ESN are: the reservoir size, the spectral radio, the density of connection, the input and output scale, the input and output shift, and the activation function (Verstraeten et al., 2007, Ishii et al., 2004). Search ESN parameters using optimization algorithms was applied by Ishii et al. (2004), Ferreira, Ludermir, Aquino, Lira, Neto (2008), and Ferreira and Ludermir (2009). Usually, the search for those parameters is carried out in an exhausting way or through random experiments, which in general takes long time to be accomplished and demand high computational resource.

4.2. Proposed Approach

The RUL prognostic approach presented in this paper used historical condition monitoring dataset and event data (Run-to-Failure information) of a group of machines, equipment or system with similar characteristics. The proposed architecture is shown in the Figure 2, and include three modules: Data Acquisition, Optimization and Training, and RUL Prognostic.

Data Acquisition: This module collects condition monitoring data and event data from sensors installed on machines. Signal processing can be applied when necessary to the data collected in order to extract and select relevant characteristics for prognostic. The total dataset needs to be separated into training and testing dataset.

Solution: The approach developed focus to optimize the dynamic reservoir parameter, specific six of them and generate the matrix weight W randomly. The parameters are: the reservoir size, the spectral radio, the input and output scale, and the input and output shift. The solution S is represented through a vector format showed in the Eq. (8).

$$S = [N, sr, IS, IF, OS, OF] \quad (8)$$

Where: N is a reservoir size, sr is the spectral radio, IS is the input scale, IF is the input shift, OS is the output scale, and OF is the output shift.

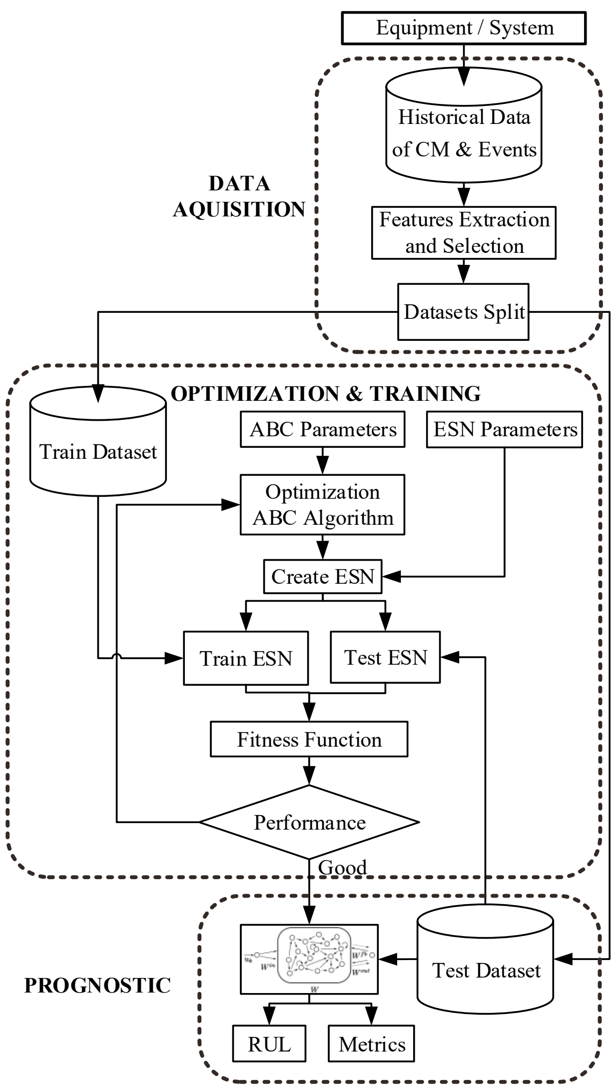

Figure 2. ESN-ABC Architecture

Fitness Function: Inspired in the publication of Ferreira, Ludermir, and Aquino (2013), in this paper is used a fitness function showed in the Eq. (9), which tries to play the GL criterion presented in Proben1 (Prechelt, 1994). The fitness function is based on the performance in the training set and in the test set, choosing this function minimizes the chances of overfitting.

$$f = NRMSE_{train} + |NRMSE_{train} - NRMSE_{test}| \quad (9)$$

Where f is the value to be minimized by the ABC algorithm and the NRMSE (Normalized Root Mean-square Error) is calculated as in Eq. (10), where $NRMSE_{train}$ is the average of NRMSE in the training set and $NRMSE_{test}$ is the average of NRMSE in the test set.

$$NRMSE = \frac{1}{LP}\sum_{i=1}^{P}\sum_{j=1}^{L}\sqrt{\frac{\left(y_{ij} - y_{targetij}\right)^2}{var(y_d)}} \quad (10)$$

Where: P is the total number of patterns in the set, L is the number of output units of the ESN, y_{ij} and $y_{targetij}$ are actual and desired outputs of the i^{th} neuron in the output layer, respectively.

Parameters: The application of the ABC algorithm along with ESN to failure prognostic is important to choose the parameters that have influence in its performance. The parameters must be chosen according the problem necessity and the resource available. In the Table 1 and Table 2 is detailed the main parameters of the ABC algorithm and the ESN.

Parameter	Description	Value
COL	Colony Size (employed+onlooker Bees)	[100, 1000]
BN	Initial Employed Bees	COL/2
SN	Food Source Number	BN
BC	Initial Onlooker Bees	COL-BN
maxTrial	A food source that can't be improved	[50, 500]
maxIter	Cycles number	[10, 100]
D	Number of parameters to be optimized	Size (S)
f	The fitness function to be minimized	Eq. (9)
ub	Upper limit of the parameters	max(S)
lb	Lower limit of the parameters	min(S)

Table 1. ABC algorithm parameters

ESN-ABC Algorithm: The search process of the ABC algorithm consist of a step sequence where a set of solutions passes through the selection process. This process is divided in three phases described in the Algorithm 1: the inputs that describe the parameters, the optimization process where is realizes the tuning of parameters minimizing the fitness function as detailed in the Algorithm 2, and finally, the results show the better ESN parameters.

Parameter	Description	Value
NE	Input number of the ESN	# variables
N	Reservoir size	[20, 500]
IS	Input Scale	[0.01, 1]
IF	Input Shift	[-1, 1]
OS	Output Scale	[0.01, 1]
OF	Output Shift	[-1, 1]
sr	Spectral Radio	[0.01, 1]
c	Reservoir connectivity	[0.1, 0.5]

Table 2. ESN parameters

Algorithm 1: Pseudo-code for ESN-ABC Algorithm

```
Input: SN, S, f, [xmin; xmax], maxTrial, maxIter
Output: Better Solution Sg: ESN parameters and weights
Start
    Generate random position for the SN Food source
    Eq. (4) and calculate fitness f(S)
    Repeat
      // EMPLOYED BEE PHASE
      For each solution i calculate the neighborhood k
       and dimension j
       Produce a new solution using the Eq. (6)
       Calculate the fitness value f(S)
       Update positions if f(Si) improve the last value
       //ONLOOKER BEE PHASE
       Calculate the probabilities vector pi using Eq. (7)
       For i = 1 : SN do
          If rand() > pi then
             Determine a neighborhood k and dimension j
            Produce a new solution Si using the Eq. (4)
            Calculate the fitness value f(Si)
            Update positions if f(Si) improve the last value
         End If
       End For
       //SCOUT BEE PHASE
       Determine the abandoned solution and send scout
       bee to search new food source
       Update the best solution Sg Acording the fitness
       iter = iter + 1
    Until iter <= maxIter
End
```

4.3. RUL Prognostic

The RUL prognostic approach is based on historical condition monitoring data. An ESN with parameters defined by the ABC algorithm realizes the prognostic process. After the training process, the ESN with their parameters adjusted and weight trained will be capable to estimate RUL. The RUL prognostic is perform processing the test dataset. The prognostic result is compared with the true RUL present in the test data. The results achieved by the ESN-ABC is compared to the results of other researcher through prognostic metrics, this metrics result of a mathematical equations having as a input variables the estimated RUL and the true RUL.

Algorithm 2: Fitness function pseudo-code

```
Input: Train Dataset, Test Dataset, ESN Parameters
Output: Fitness function value: f
1: Start
2:         Load Train Dataset
3:         Load Test Dataset
4:         Create ESN with ESN Parameters
5:         Training the readout weights
6:         Calculate train dataset error Eq. (10)
7:         Calculate test dataset error Eq. (10)
8:         Calculate the fitness funtion value Eq. (9)
9: End
```

5. CASE STUDY: TURBOFAN ENGINE

The result of the RUL prognostic algorithm based on ESN-ABC is demonstrated through a case study of turbofan engines (Figure 3) from the NASA Prognostic Data Repository (Saxena & Goebel, 2008). The structure of the dataset is described and then the effectiveness of the proposed is demonstrated and the results compared with other results.

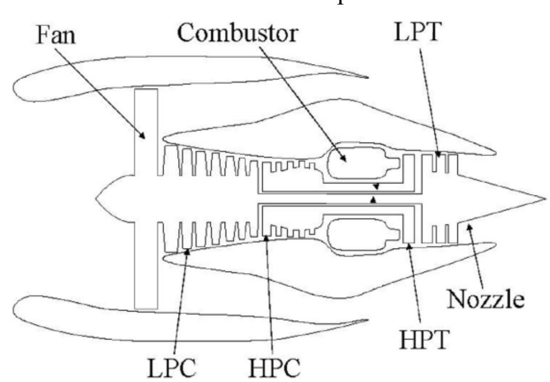

Figure 3. Turbofan engine

5.1. Turbofan Data

The prognostic dataset is a result of run-to-failure experiments simulated to investigate the degradation of the turbofan engine system. The simulation model was built on Commercial Modular Aero-Propulsion System Simulation that was developed at NASA Army Research Laboratory (Frederick, De Castro, and Litt, 2007).The repository has four datasets generated from independent simulation experiments. The datasets consist of multi-variate time series signals from different degrading instances and contaminated with noise. Each dataset contains engine units that are in the same manufacturing batch, but with a different initial state.

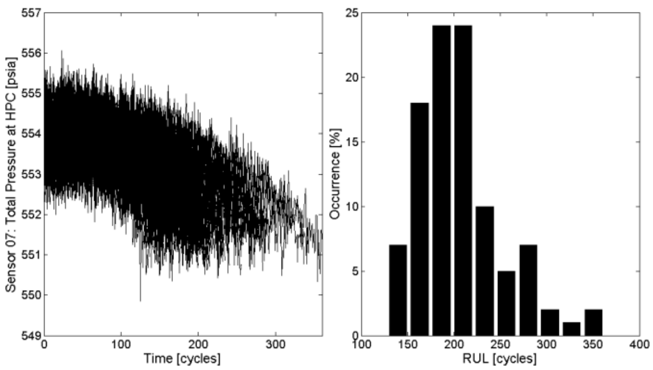

Figure 4. Sensor measurement and RUL distribution

Each engine begins from a normal operation, but due to some fault occurrence, starts to degrade. The fault magnitude increases with time until functional failure takes place. The first dataset has 100 trajectories for train and 100 for test, one operating conditions (sea level), and one fault mode is the HPC (High-Pressure Compressor) degradation. The second dataset has 260 trajectories for train and 259 for test, six operational conditions, and one fault mode (HPC degradation). The third dataset has 100 trajectories for train and 100 for test, one operating conditions (sea level), and two fault modes (HPC degradation, Fan Degradation). The fourth dataset has 248 trajectories for train and 249 for test, six operational conditions, and two fault modes (HPC degradation, Fan Degradation).

Each unit is further divided in “training” and “test” subsets. The training subset contain examples of units that run until failure, while the test subset end sometime before to failure. In this work is used the first dataset “train_FD001.txt” composed of 100 training engines (with different temporal length or life as shown in the Figure 4), and “test_FD001.txt” also with 100 engines. It should be noted that the test data are composed of pieces of trajectories and remaining life is unknown.

Each cycle either to train or for test, contains 24 dimensional time series (3 operating conditions and 21 sensor measurements). From the 21 condition measurements, only 14 are used in this paper based on the results obtained by Wang (2010), these fourteen signals are from sensors {2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, 21} that present tendency and provide the degradation trajectory to perform prognostic. Furthermore, the selection of variables reduces the dimensionality of the problem.

5.2. Prognostic Results

In this section is described the configuration parameters, the topology and the results obtained by classical ESN and the hybrid approach ESN-ABC developed in this work. The ESN and ESN-ABC algorithm were developed in Matlab based on ToolboxESN (2015) and ToolboxABC (2015) to implement the ESN and the optimization ABC algorithm, respectively. The experiments conducted in this study are divided into two subsections. First, implemented a classic ESN where the parameters are defined manually. The second part of this section deals with the proposed approach, adjusting the optimal parameters of the ESN through the ABC algorithm.

Parameters	Scale	Step	Value
N	[40, 300]	5	150
sr	[0.01, 1]	0.05	0.5
IS	[0.01, 1]	0.05	0.05
IF	[-1, 1]	0.05	0.95
OS	[0.01, 1]	0.05	0.005
OF	[-1, 1]	0.05	-0.05

Table 3. Parameters setting for classical ESN

Classical ESN Approach: The RUL prognostic of turbofan engines is realized through a classical ESN. For this purpose is necessary condition monitoring signals and a continue variable that represents the number of cycles remaining to the failure occurrence. This variable represents the RUL and considered as a target for ESN training. The result obtained by the classical ESN is useful as a reference for the ESN-ABC model. A practical guide for ESN application published by Lukosevicius (2012) was used as a reference to set the ESN parameters as shown in the Table 3. The estimated RUL by the classical ESN and the true RUL, and the RUL residual for 100 engines of the test subset is shown in the Figure 5. As can be observed there are 24% of the estimated RUL between the false positive and false negative threshold.

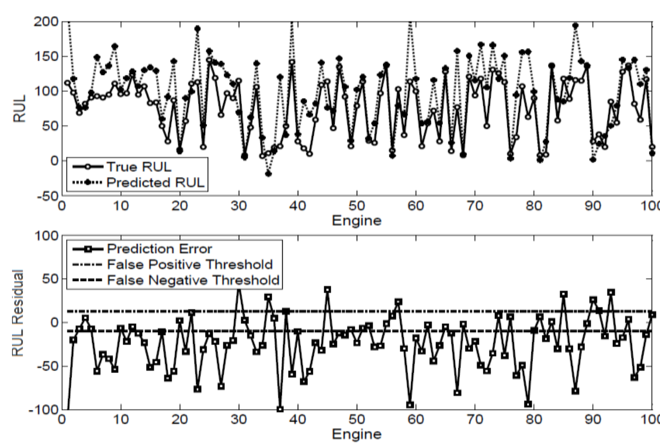

Figure 5. Classical ESN trained

ESN-ABC Approach: This approach starts generating the input weights W^{in} as a random uniform distribution. Next, is initialized the parameters, defined reservoir weights randomly. The ABC algorithm starts the iterative process, adjusting the parameters in order to minimize the fitness function (Eq. (9)). Finally is calculated the readout weights. The parameters founded experimentally by the ABC algorithm is shown in the Table 4.

N	sr	IS	IF	OS	OF
198	0.09	0.83	0.38	0.14	0.83

Table 4. Parameters Setting by ABC Algorithm

The Figure 6 shows the estimated and the true RUL by the ESB-ABC approach, and the RUL residual for 100 engines of the test dataset. Can be observed that the ESN-ABC approach obtain good results for RUL prognostic. The RUL residual (34%) are between a false positive and false negative threshold, +13 and -10 respectively, these thresholds was defined by Saxena & Goebel (2008).

5.3. Performance Metrics

Determining precision, accuracy and performance of prognostic algorithm is a recent topic. The taxonomy of these performance metrics for RUL estimation was proposed by Saxena, Celaya, Balaban, Goebel, Saha, Saha, and Schwabacher (2008a) and Saxena, Celaya, Saha, Saha, and Goebel (2010) where was presented different categories based on: accuracy, precision, and specifically for prognostic (PHM metrics). The Table 5 shows publications that use metrics for dataset 1.

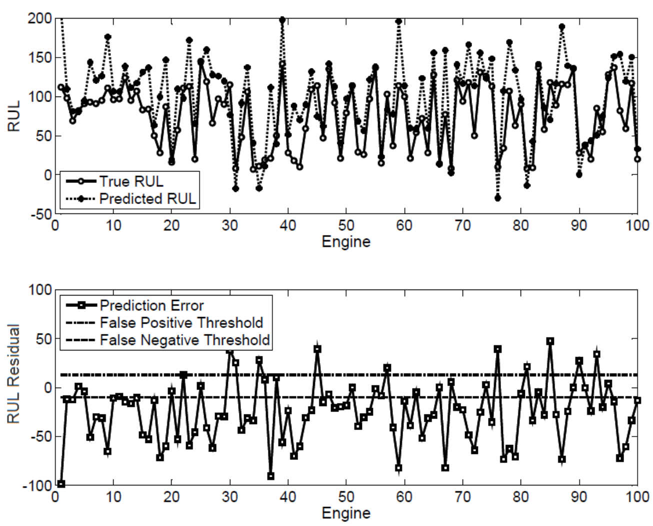

Figure 6. ESN trained by ABC algorithm

In order to assess the performance of the case study, it is realized the comparison between the estimated RUL and the true RUL "rul_FD001.txt". Using the prognostic metrics equation presented by Saxena et al. (2010) is obtained the prognostic metric values that are used to perform a quantitative comparison. The metric PHM08 have been used in the PHM08 competition presented by Saxena, Goebel, Simon and Eklund (2008b) is described by the Eq. (11).

$$PHM08 = \begin{cases} \sum_{m=1}^{M} e^{-\left(\frac{\varepsilon_m}{a_1}\right)} - 1 \quad para \quad \varepsilon < 0 \\ \sum_{m=1}^{M} e^{\left(\frac{\varepsilon_m}{a_2}\right)} - 1 \quad para \quad \varepsilon \geq 0 \end{cases} \tag{11}$$

Where: a_1 and a_2 are the parameters that control the asymmetric preference. ε_m is a RUL residual.

The FPN (False Positive Number) is calculated by Eq. (12) using the thresholds defined in the PHM08.

$$FPN = \begin{cases} 1 & \varepsilon > t_{FP} \\ 0 & Other \end{cases} \tag{12}$$

Where: ε is a RUL residual and t_{FP} is the false positive threshold.

The Eq. (13) describe MSE (Mean Square Error), the Eq. (14) the MAE (Mean Absolute Error). The ME (Mean Error) represented by the Eq. (15), the MAD (Mean Absolute Deviation) by the Eq. (16). The MAPE (Mean Absolute Percentage Error) described in the Eq. (17), this metric quantifies the error in percentage.

$$MSE = \frac{1}{M}\sum_{m=1}^{M} \varepsilon_m^2 \tag{13}$$

$$MAE = \frac{1}{M}\sum_{m=1}^{M} |\varepsilon_m| \tag{14}$$

$$ME = \frac{1}{M}\sum_{m=1}^{M} \varepsilon_m \tag{15}$$

$$MAD = \frac{1}{M}\sum_{m=1}^{M}|\varepsilon_m - \bar{m}|, \bar{m} = median(\varepsilon_m) \tag{16}$$

$$MAPE = \frac{1}{M}\sum_{m=1}^{M} \left|\frac{100\varepsilon_m}{t_{RUL}}\right| \tag{17}$$

Where: ε is a RUL residual and M is the total engine numbers, and t_{RUL} is the true RUL of the engine *m*.

The metrics calculated is compared with the metrics obtained by other researchers that used the same dataset of the NASA prognostic data repository, the results are shown in the Table 5.

Comparison based on PHM8 metric show that ESN-ABC approach presents better result that classical ESN. The approach presented by Peng, Wang, Wang, Liu, and Peng (2012) based on classical ESN obtain as a prognostic metric MSE = 3969. In this paper the classical ESN obtains MSE=1558 and the ESN-ABC obtains MSE = 1415. The difference in the results can be because Peng et al. (2012) use all the 24 dimension time series, and ESN parameters *N*=90, *sr* = 0.05, in this work were used only 14 sensor measurements, *N*=150 and *sr* = 0.5. The approach overcome the results obtained in term of mean square error. Analyzing the other prognostic metrics can be observe that ESN trained by ABC algorithm, though consume more computational resource, have better performance than the classical ESN trained manually.

Liu, Gebraeel and Shi (2013) present a data-level approach that use composite health index using in their case study the first dataset of the prognostic repository. They use 11 sensor measurement and calculate the metric MAPE for each sensor selected. The MAPE value is better that the ESN-ABC.

	Accuracy					Precision	
	PHM08	**FPN**	**MSE**	**MAPE**	**MAE**	**ME**	**MAD**
ESN-ABC	7634	14	1415	39.5	28.8	21.4	22.6
ESN	9988	10	1558	63.9	31.5	24.1	24.3
Peng (2012)	---	---	3969	---	---	---	---
Liu et al. (2013)	---	---	---	9	---	---	---

Table 5. Metrics Comparison for train/test dataset

6. CONCLUSION

In this paper is presented a hybrid approach that uses the ABC algorithm for setting ESN parameters, this solution is applied to RUL prognostic. ESN is an efficient technique to design and train a RNN. On the other hand, the ABC algorithm has been successfully used for optimization problems.

Recently, several approaches have been presented for design and train the dynamic reservoir. As a contribution, we use the ABC algorithm for adjusting a subset of the reservoir parameters. Setting the parameters of the reservoir using meta-heuristics can be a very expensive task. Therefore, setting ESN parameter is performed using the ABC algorithm in an automatic way. However, the proposed approach demands a high computational cost due to the large search space, especially for higher values of N.

The ESN-ABC source code implemented in Matlab is available in https://sourceforge.net/projects/esn-abc/, also on this site is attached the tutorial video in Portuguese. The application development organized into three modules: data acquisition, training, and RUL prognostic, according to the proposed approach. The software application is a friendly GUI (Graphic User Interface) tool, with the objective to test the ESN-ABC approach.

The possibility to get the best ESN parameters is one of the main advantages of this proposed approach.

For future work, we are working to test the approach to the others 3 dataset of the prognostic repository where will be possible compare with researchers that used these datasets. Another research field identified is extend the same procedure to other Reservoir Computing methods such as Liquid State Machines (LSM) and Backpropagation Decorrelation. As well as, it is interesting to implement and compare the performance reached by the ABC algorithm with other swarms intelligent and bio-inspired techniques.

ACKNOWLEDGEMENT

The authors acknowledge to the PPMEC (Mechatronic Post-graduation Program) at the University of Brasilia, for the technical support.

NOMENCLATURE

ABC	Artificial Bee Colony
ANN	Artificial Neural Network
BPDC	Backpropagation Decorrelation
CBM	Condition Based Maintenance
DR	Dynamic Reservoir
EoL	End of Life
ESN	Echo State Network
ESP	Echo State Property
DE	Differential Evolution
EKF	Extended Kalman Filter
FPN	False Positive Number
GA	Genetic Algorithm
HPC	High-Pressure Compressor
KF	Kalman Filter
LSM	Liquid State Machines
MLP	MultiLayer Perceptron
MSE	Mean Square Error
MAD	Mean Absolute Deviation
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ME	Mean Error
NRMSE	Normalized Root Mean-square Error
PHM	Prognostic and Health Management
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
RC	Reservoir Computing
RLS	Recursive Least Square
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SN	Solution Number

ACKNOWLEDGMENT

The authors acknowledge to the GRACO (Automation and Control Group) at the University of Brasilia (UnB) and to the NASA for the dataset prognostic repository to validate the proposed approach. The authors also thank the anonymous reviewers and to the editor for their valuable comments

REFERENCES

Boccato, L. (2013). *Novas propostas e aplicações de redes neurais com estados de eco*. Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, Brasil.

Bush, K., & Tsendjav, B. (2005). Improving the richness of echo state features using next ascent local search. *In Proceedings of the artificial neural networks in engineering conference* (pp. 227–232).

Butani, R.C., Gajjar, B.D. and Thakker, R.A. (2011). Performance evaluation of Particle SwarmOptimization (PSO) and Artificial Bee Colony (ABC) Algorithm. *International Conference on Advanced Computing, Communication and Networks*.

Compare, M. and Zio, E. (2014). Predictive Maintenance by Risk Sensitive Particle Filtering. *Reliability IEEE Transactions on*, vol.63, no.1, pp.134-143.

Daroogheh, N., Meskin, N. and Khorasani, K. (2014). A novel particle filter parameter prediction scheme for failure prognosis. *American Control Conference (ACC)*, vol., no., pp.1735-1742.

Dong, M. and He, D., (2007). Hidden semi-Markov model-based methodology for multisensory equipment health diagnosis and prognosis. *European Journal of Operational Research* 178 (3), 858–878.

Ferreira, A. A., Ludermir, T.B., Aquino, R., Lira, M.M. and Neto, O.N. (2008). Investigating the use of reservoir computing for forecasting the hourly wind speed in short-term. *In International Joint Conference on Neural Networks - IJCNN*, pages 1950–1957, Hong Kong.

Ferreira, A. A. e Ludermir, T.B. (2009). Genetic algorithm for reservoir computing optimization. *Neural Networks. IJCNN - International Joint Conference on*, vol., no., pp.811-815.

Ferreira, A. A., Ludermir, T.B. and Aquino, R. (2013). An approach to reservoir computing design and training. *Expert Systems with Applications*, Volume 40, Issue 10, pp. 4172-4182.

Ferreiro, S., Arnaiz, A., Sierra, B. and Irigoien, I. (2012). Application of Bayesian networks in prognostics for a new Integrated Vehicle Health Management concept. *Expert Systems with Applications*, Volume 39, Issue 7, Pages 6402-6418.

Frederick, D., De Castro, J. and Litt, J. (2007). “User’s Guide for the Commercial Modular Aero-Propulsion System Simulation (CMAPSS).” NASA/ARL.

Heng, A., Zhang, S. Tan, A. and Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, Volume 23, Issue 3, pp. 724-739.

Hu, C., Youn, B.D. and Wang, P. (2011). Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Prognostics and Health Management (PHM)*, IEEE Conference on , vol., no., pp.1-10.

Hossain, M.S. and El-Shafie, A. (2014). Evolutionary techniques versus swarm intelligences: application in reservoir release optimization. *Neural Computing and Applications*, volume 24, number 7-8, pp.1583-1594.

Gasperin, M., Juricic, Baskoski, P. and Jozef, V. (2011). Model-based prognostics of gear health using stochastic dynamic models. *Mechanical Systems and Signal Processing* 25, pp. 537–538.

Ishii, K., van der Zant, T., Becanovic, V., & Ploger, P. (2004). Identification of motion with echo state network. *In Proceedings of the OCEANS MTS/IEEE–TECHNOOCEAN conference*, Vol. 3, pp. 1205–1210.

Jaeger, H (2001). *The Echo state approach to analyzing and training recurrent neural networks*. Technical Report GMD Report 148, German National Research Center for Information Technology, URL www.faculty.jacobs-university.de/hjaeger/pubs/ EchoStatesTechRep.pdf.

Jaeger, H (2002a). *Short-term memory in echo state networks*. Technical report, GDM 152, German National Resource Center for Information Technology.

Jaeger, H (2002b). *A tutorial on training recurrent neural networks*, covering BPTT, RTRL, EKF and the echo state network approach. Technical Report GMD Report 159, German National Research Center for Information Technology, URL www.faculty.jacobs-university.de/hjaeger/pubs/ESNTutorial Rev.pdf.

Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks, *in Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, K. Obermayer, Eds., Cambridge, MA: MIT Press, pp.593-600.

Jaeger, H. and Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667): pp. 78-80, URL www.faculty.jacobs-university.de/ hjaeger/pubs/ ESNScience04.pdf.

Karaboga, D. and Basturk, B. (2007). A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *Journal of Global Optimization*, Volume: 39, Issue: 3, pp. 459-171.

Karaboga, D. and Akay, B. (2009). A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation*, Volume 214, Issue 1, Pages 108-132.

Karaboga, D. and Ozturk, C. (2009). Neural Networks Training by Artificial Bee Colony Algorithm on Pattern Classification. *Neural Network World*, 19 (3), 279-292.

Karaboga, D. and Ozturk, C. (2010). Fuzzy clustering with artificial bee colony algorithm. *Scientific Research and Essays*, Volume: 5 Issue: 14 Pages: 1899-1902.

Karaboga, D. (2010). *Artificial bee colony algorithm*. Scholarpedia, 5(3):6915.

Kumar, S., Torres, M., Chan, Y.C. and Pecht, M. (2008). A hybrid prognostics methodology for electronic products. *Neural Networks. IJCNN*. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, vol., no., pp. 3479-3485.

Li, D., Wang, W., and Ismail, F. (2013). Enhanced fuzzy-filtered neural networks for material fatigue prognosis. *Applied Soft Computing*, Vol. 13, No. 1, pp. 283–291.

Liao, L. and Kottig, F. (2014). Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction. *Reliability, IEEE Transactions on*, vol.63, no.1, pp. 191-207.

Liu, K., Gebraeel, N. Z. and Shi, J. (2013). A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE Trans. on Automation Science and Engineering*.

Lukosevicius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3, pp.127–149.

Maass, W., Natschlager, T. and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560.

Natschlager, T., Maass, W. and Markram, H. (2002). The 'liquid computer': A novel strategy for real-time computing on time series. *Special Issue on Foundations of Information Processing of TELEMATIK*, 8(1):39–43.

Pecht, M. (2008). *Prognostics and Health Management of Electronics*. John Wiley, New Jersey.

Pecht, M. and Jaai, R. (2010). A prognostics and health management roadmap for information and electronics-rich system. *Microelectronics Reliability* 50, pp. 317–323.

Peng, Y., Wang, H., Wang, J., Liu, D. and Peng, X. (2012). A modified echo state network based remaining useful life estimation approach. *Prognostics and Health Management (PHM), IEEE Conference on*, vol., no., pp.1-7.

Pla, A., López, B., Gay, P. and Pous, C. (2013). eXiT*CBR.v2: Distributed case-based reasoning tool for medical prognosis, *Decision Support Systems*, Volume 54, Issue 3, pp. 1499-1510.

Prechelt. L. (1994). *Proben1 - a set of neural network benchmark problems and benchmarking rules*. Technical report, 21/94, Fakultät für Informatik, Universität Karlsruhe, Germany.

Ramasso, E. and Saxena, A. (2014). Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets," International Journal of Prognostics and Health Management, v. 5, n. 2, p. 1–15.

Saxena, A. and Goebel, K. (2008). *Turbofan Engine Degradation Simulation Data Set*, NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S. and Schwabacher, M. (2008a). *Metrics for Evaluating Performance of Prognositc Techniques*. International Conference on Prognostics and Health Management, Denver, CO.

Saxena, A., Goebel, K., Simon, D. and Eklund, N. (2008b). *Damage propagation modeling for aircraft engine run-to-failure simulation*. In Proceedings of the 2008 International Conference on Prognostics and Health Management, p. 1–9. International Conference On Prognostics And Health Management, Denver, CO, Oct 06-09.

Saxena, A., Celaya, J., Saha, B., Saha, S. and Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*.

Schrauwen, B., Verstraeten, D. and Campenhout, J.V. (2007). An overview of reservoir computing: theory, applications and implementations. *In Proceedings of the 15th European Symposium on Artificial Neural Networks*, pp. 471–482.

Shankar, B. (2015). *Remaining Useful Life Prediction through Failure Probability Computation for Condition-based Prognostics*. In Proceedings of the Annual Conference of the Prognostics and Health Management Society.

Song, Q., Zhao, X. Feng, Z. An, Y. and Song, B. (2011). Hourly electric load forecasting algorithm based on echo state neural network, *Control and Decision Conference (CCDC)*, 2011 Chinese, vol., no., pp.3893, 3897, 23-25 May.

Steil, J.J. (2004). Backpropagation-decorrelation: online recurrent learning with O(N) *complexity. In Proc. IJCNN*.

ToolboxABC (2015). *MATLAB code v2 of the basic ABC algorithm*. Available in: < http://mf.erciyes.edu.tr/abc>. Access in: march 2015.

ToolboxESN (2015). *Simple and very simple Matlab toolbox for Echo State Networks*. Available in: <http://reservoir-computing.org>. Access in: march 2015.

Turanoglu, E., Ozceylan, E. and Kiran, M.S. (2011). Particle Swarm Optimization and Artificial Bee Colony Approaches to Optimize of Single Input-Output Fuzzy Membership Functions. *Proceedings of the 41st International Conference on Computers & Industrial Engineering*.

Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A. and Wu, B. (2006). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. 1st ed. Hoboken, New Jersey: John Wiley & Sons, Inc.

Verstraeten, D., Schrauwen, B., D'Haene, M. and Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, 20:391–403.

Wang, T., (2010). *Trajectory Similarity Based Prediction for Remaining Useful Life Estimation*. Doctoral dissertation. University of Cincinnati.

Wang, W., Zhang, W., (2008). An asset residual life prediction model based on expert judgments. *European Journal of Operational Research* 188, pp. 496–505.

Weiming, W., Bing, L., Min, L. and Houjun, W. (2014). Prognostics of Lithium-Ion Batteries Based on the Verhulst Model, Particle Swarm Optimization and Particle Filter. *Instrumentation and Measurement, IEEE Transactions on*, vol.63, no.1, pp.2-17.

Zhang, Y., Zhao, X., Liu, W., Zhang, J., Jia, Y. and Feng, T. (2011). Research on gearbox wearing prognosis based on Gamma-State Space Model. *Reliability, Maintainability and Safety (ICRMS), 9th International Conference on*, vol., no., pp. 279-283.

BIOGRAPHIES

Edgar Amaya received his PhD in Mechatronics Systems from the University of Brasilia (UnB), Brazil, in 2015, his M.Sc. degree in Mechatronics Systems from the UnB in 2008. Moreover, his B.S. degree in Mechatronic Engineering from the University of engineering (UNI), Peru, in 2002. His research interests include condition monitoring, real-time expert systems for diagnostics, neural networks for prognostics, and intelligent system applications in power engineering. His current research focuses on artificial intelligence for monitoring, diagnosis and prognosis machine's reliability.

Alberto Alvares is a Professor of the Mechanical Engineering department of the University of Brasilia (UnB). He received a bachelor's degree in Mechanic Engineering from UnB, Brazil, 1986, a M.D. degree in Mechanic Engineering from UFSC (Universidade Federal de Santa Catarina), Brazil, 1990, and the D.S. degree in Mechanic Engineering from the UFSC, (2005)